Algorithmics Exam

Question 1: Find similar patterns of colors across multiple images

Answer:

Paper: FINDING COLOR AND SHAPE PATTERNS IN IMAGES by Scott Cohen

Book: NI Vision Concepts Help

Article: **Exploring Image Similarity Approaches in Python (Medium) by Vasista Reddy, Search-by-Colour (Medium) by Joshua Comeau, The Microsoft Windows Palette Manager(compuphase)**

AI: ChatGbt (To fix the code bugs, to enhance the performance, understanding code logic)

Image similarity can be thought of as a numerical representation of how alike two images are in terms of their visual content. There are several dimensions along which images can be similar, such as color, shape, texture, and composition.

I have used the following methods to identify the similar patterns of colors across multiple images,

1. Histogram Based Approach
2. Structural Similarity Index (SSIM)
3. Color Feature Extraction - Dominant Colors

Histogram Based Approach:

The idea behind this approach is to get the histogram of the given images and compare it with each other to identify their similarities and errors. The method is based on fitting the histogram of the measured image to the histogram of a model function, and it can be used for contrast determination in fringe patterns. Simulated and experimental results are presented.

- A histogram represents the distribution of pixel intensity values for an image (e.g., for each color channel like red, green, blue).

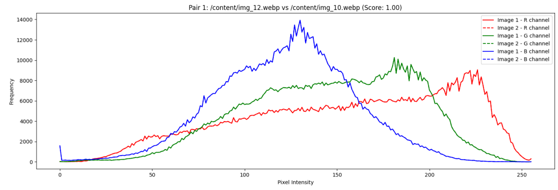
- This method computes histograms for two images and compares them using metrics like correlation, intersection, or chi-square.
- It captures the overall distribution of colors in the images, ignoring spatial layout.

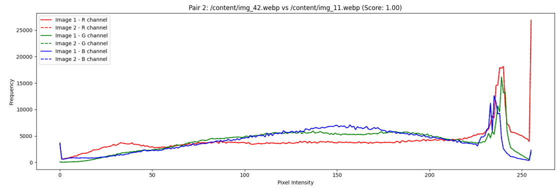
Strengths:

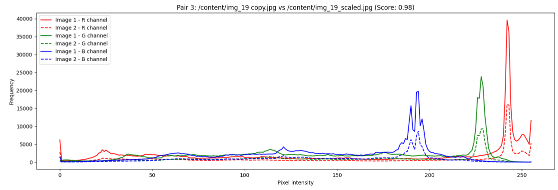
- **Robustness**: Works well for global comparisons; insensitive to minor spatial variations or transformations like rotation or translation.
- **Computational Simplicity**: Easy to implement and compute, making it suitable for large datasets.
- **Universal Application**: Does not require images to be the same size or resolution.

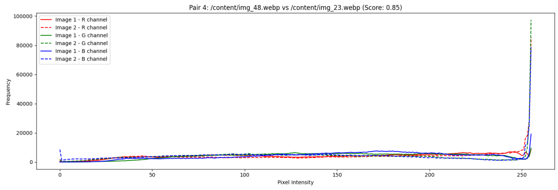
Weaknesses:

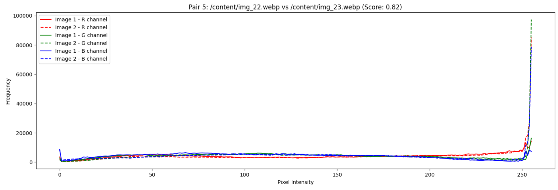
- **Spatial Insensitivity**: Ignores the spatial arrangement of colors. Two images with identical color distributions but different compositions can have identical histograms.
- **Lighting Sensitivity**: Results can vary significantly with changes in brightness or contrast.

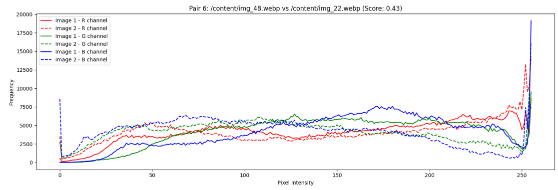

Pair 1: /content/img_12.webp vs /content/img_10.webp (Score: 1.00)
Image 1 - R channel
Image 2 - R channel
Image 1 - G channel
Image 2 - G channel
Image 1 - B channel
Image 2 - B channel
Frequency
Pixel Intensity

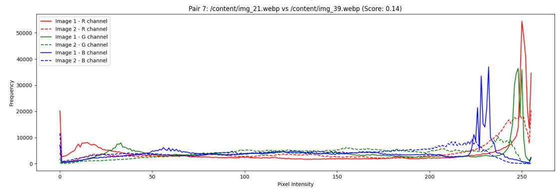

Pair 2: /content/img_42.webp vs /content/img_11.webp (Score: 1.00)
Image 1 - R channel
Image 2 - R channel
Image 1 - G channel
Image 2 - G channel
Image 1 - B channel
Image 2 - B channel
Frequency
Pixel Intensity

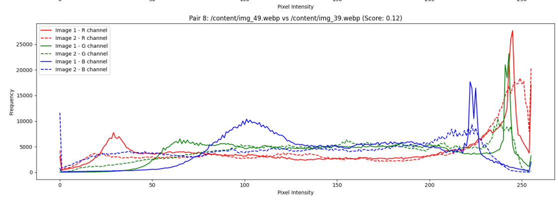

Pair 3: /content/img_19 copy.jpg vs /content/img_19_scaled.jpg (Score: 0.98)
Image 1 - R channel
Image 2 - R channel
Image 1 - G channel
Image 2 - G channel
Image 1 - B channel
Image 2 - B channel
Frequency
Pixel Intensity

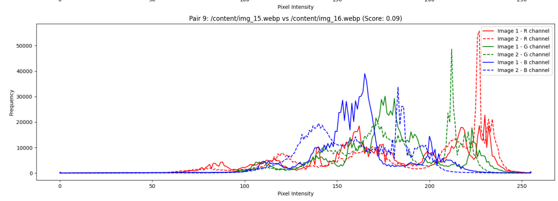

Pair 4: /content/img_48.webp vs /content/img_23.webp (Score: 0.85)
Image 1 - R channel
Image 2 - R channel
Image 1 - G channel
Image 2 - G channel
Image 1 - B channel
Image 2 - B channel
Frequency
Pixel Intensity

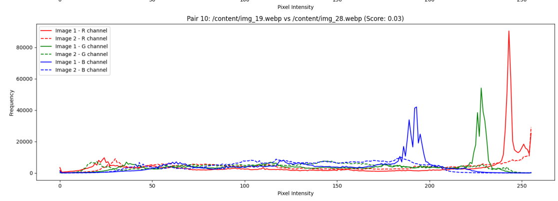

Pair 5: /content/img_22.webp vs /content/img_23.webp (Score: 0.82)
Image 1 - R channel
Image 2 - R channel
Image 1 - G channel
Image 2 - G channel
Image 1 - B channel
Image 2 - B channel
Frequency
Pixel Intensity


Pair 6: /content/img_48.webp vs /content/img_22.webp (Score: 0.43)
Image 1 - R channel
Image 2 - R channel
Image 1 - G channel
Image 2 - G channel
Image 1 - B channel
Image 2 - B channel
Frequency
Pixel Intensity


Pair 7: /content/img_21.webp vs /content/img_39.webp (Score: 0.14)
Image 1 - R channel
Image 2 - R channel
Image 1 - G channel
Image 2 - G channel
Image 1 - B channel
Image 2 - B channel
Frequency
Pixel Intensity


Pair 8: /content/img_49.webp vs /content/img_39.webp (Score: 0.12)
Image 1 - R channel
Image 2 - R channel
Image 1 - G channel
Image 2 - G channel
Image 1 - B channel
Image 2 - B channel
Frequency
Pixel Intensity


Pair 9: /content/img_15.webp vs /content/img_16.webp (Score: 0.09)
Image 1 - R channel
Image 2 - R channel
Image 1 - G channel
Image 2 - G channel
Image 1 - B channel
Image 2 - B channel
Frequency
Pixel Intensity


Pair 10: /content/img_19.webp vs /content/img_28.webp (Score: 0.03)
Image 1 - R channel
Image 2 - R channel
Image 1 - G channel
Image 2 - G channel
Image 1 - B channel
Image 2 - B channel
Frequency
Pixel Intensity

Color Moments - Dominant Colors (K-Means Clustering)

- K-Means clustering groups similar pixels in an image based on their RGB values, identifying a set of dominant colors.
- The image is reduced to a compact representation based on these dominant colors.
- The similarity is computed using distances between the clusters of two images (e.g., Euclidean distance).

Strengths:

- **Compact Representation**: Summarizes the image in terms of its most dominant colors, reducing noise and irrelevant details.
- **Effective for Visual Themes**: Works well for images with distinctive color palettes, such as sunsets or forests.
- **Flexible**: The number of clusters (`k`) can be adjusted to control the level of abstraction.

Weaknesses:

- **Loss of Detail**: Fine-grained color variations or small features may be ignored, as the focus is on dominant colors.
- **Clustering Dependency**: The effectiveness depends heavily on the choice of `k` and the initial placement of cluster centroids.
- **Color Space Sensitivity**: Differences in color space representation (e.g., RGB vs. LAB) can affect clustering outcomes.

Output:

Top 10 Similar Images (Dominant Colors):

1. Similarity between /content/img_12.webp and /content/img_10.webp: 0.0
2. Similarity between /content/img_42.webp and /content/img_11.webp: 0.0
3. Similarity between /content/img_33.webp and /content/img_41.webp: 69.75
4. Similarity between /content/img_17.webp and /content/img_19 copy.jpg: 148.11
5. Similarity between /content/img_16.webp and /content/img_48.webp: 150.22
6. Similarity between /content/img_12.webp and /content/img_44.webp: 150.37
7. Similarity between /content/img_44.webp and /content/img_10.webp: 150.37

8. Similarity between /content/img_16.webp and /content/img_45.webp: 152.1
9. Similarity between /content/img_30.webp and /content/img_19.webp: 153.63
10. Similarity between /content/img_12.webp and /content/img_16.webp: 155.05


Dominant Colors for Top Similar Image Pairs
Image 1 (Pair 1) - img_12.webp
Image 2 (Pair 1) - img_10.webp
Image 1 (Pair 2) - img_42.webp
Image 2 (Pair 2) - img_11.webp
Image 1 (Pair 3) - img_33.webp
Image 2 (Pair 3) - img_41.webp
Image 1 (Pair 4) - img_17.webp
Image 2 (Pair 4) - img_19 copy.jpg
Image 1 (Pair 5) - img_16.webp
Image 2 (Pair 5) - img_48.webp
Image 1 (Pair 6) - img_12.webp
Image 2 (Pair 6) - img_44.webp
Image 1 (Pair 7) - img_44.webp
Image 2 (Pair 7) - img_10.webp
Image 1 (Pair 8) - img_16.webp
Image 2 (Pair 8) - img_45.webp
Image 1 (Pair 9) - img_30.webp
Image 2 (Pair 9) - img_19.webp
Image 1 (Pair 10) - img_12.webp
Image 2 (Pair 10) - img_16.webp

Structural Similarity Index (SSIM)

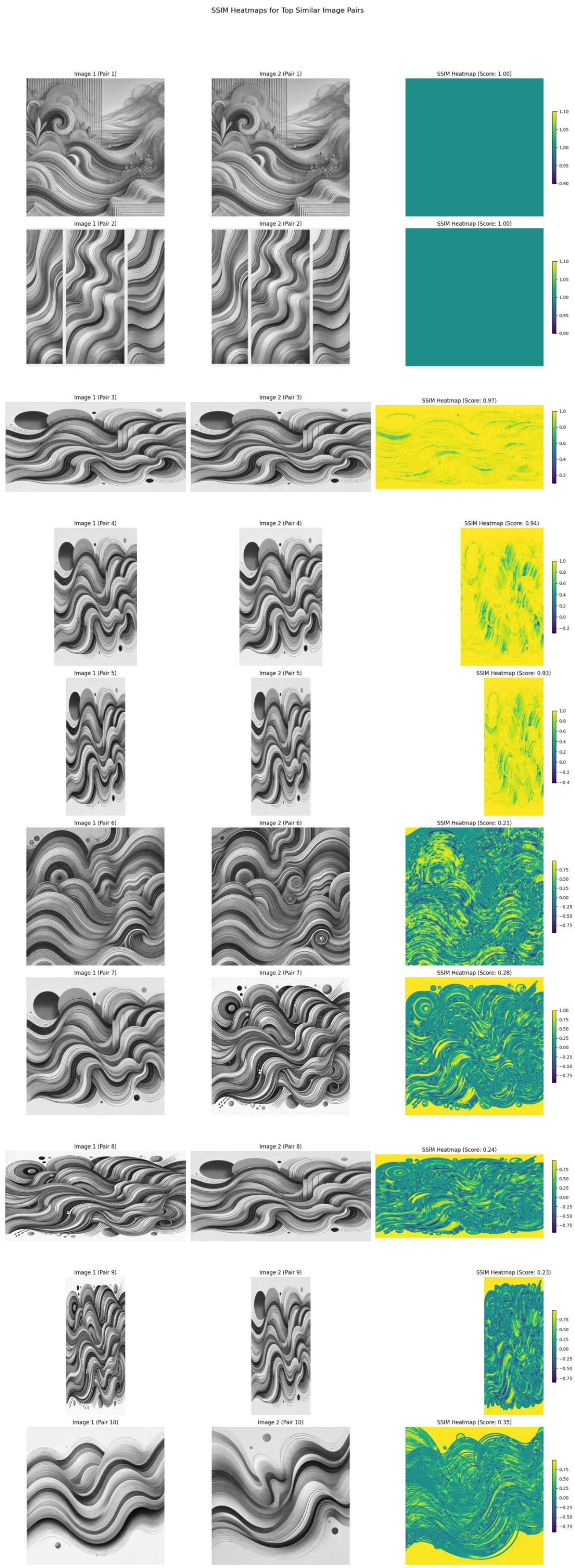
- SSIM evaluates how structurally similar two images are by comparing luminance, contrast, and spatial structure.
- It gives a similarity score between -1 and 1, where 1 indicates identical images.
- Unlike histograms or dominant color methods, SSIM focuses on perceptual quality, taking spatial relationships into account.

Strengths:

- **Perceptual Relevance**: Closely aligns with human visual perception, as it compares the structural content of images.
- **Localized Comparison**: Can highlight specific areas of similarity or dissimilarity, making it ideal for regions-of-interest comparisons.
- **Comprehensive**: Simultaneously considers multiple aspects (brightness, contrast, structure) rather than just color.

Weaknesses:

- **Same Dimension Requirement**: Both images must have identical dimensions, necessitating preprocessing like resizing or cropping.
- **Background Interference**: Differences in background content can heavily influence the similarity score.
- **Not Robust to Transformations**: Sensitive to geometric transformations (e.g., rotation, scaling).


SSIM Heatmaps for Top Similar Image Pairs
Image 1 (Pair 1)
Image 2 (Pair 1)
SSIM Heatmap (Score: 1.00)
Image 1 (Pair 2)
Image 2 (Pair 2)
SSIM Heatmap (Score: 1.00)
Image 1 (Pair 3)
Image 2 (Pair 3)
SSIM Heatmap (Score: 0.97)
Image 1 (Pair 4)
Image 2 (Pair 4)
SSIM Heatmap (Score: 0.94)
Image 1 (Pair 5)
Image 2 (Pair 5)
SSIM Heatmap (Score: 0.93)
Image 1 (Pair 6)
Image 2 (Pair 6)
SSIM Heatmap (Score: 0.21)
Image 1 (Pair 7)
Image 2 (Pair 7)
SSIM Heatmap (Score: 0.28)
Image 1 (Pair 8)
Image 2 (Pair 8)
SSIM Heatmap (Score: 0.24)
Image 1 (Pair 9)
Image 2 (Pair 9)
SSIM Heatmap (Score: 0.23)
Image 1 (Pair 10)
Image 2 (Pair 10)
SSIM Heatmap (Score: 0.35)

Method	Captures	Spatial Awareness	Robustness to Scaling/Translation	Suitability for Color Comparisons
Histogram	Global color distribution	×	✓	High
Color Moments (K-Means)	Dominant color clusters	×	✓	Medium to High
SSIM	Structural content	✓	×	Low

a)Single Vertical Line

---

1. Cosine Similarity

- **Definition**:
  - Computes the cosine of the angle between two vectors.
  - Suitable for identifying the direction of patterns (e.g., RGB distributions) rather than their magnitude.
- **Advantages**:
  - Effective for comparing color patterns across vertical lines, even if their overall brightness differs.
  - Insensitive to uniform scaling of pixel values.
- **Limitations**:
  - Less effective if there are distortions or spatial misalignments.

---

2. Dynamic Time Warping (DTW)

- **Definition**:
  - A sequence alignment method that calculates the distance between two temporal sequences, accounting for distortions.
  - Matches sequences by stretching or compressing parts to minimize distance.
- **Advantages**:

- Handles sequences with slight shifts, making it ideal for vertical lines with similar patterns but minor misalignments.
- Robust to temporal distortions (e.g., intensity shifts along the column).

- **Limitations**:
  - Computationally intensive for large datasets.
  - Sensitive to noise in pixel values.

---

3. Euclidean and Manhattan Distance

- **Euclidean Distance**:
  - Measures the straight-line distance between two vectors.
  - Computes the square root of the sum of squared differences between corresponding elements.
- **Manhattan Distance**:
  - Measures the absolute distance between two vectors.
  - Computes the sum of absolute differences between corresponding elements.
- **Advantages**:
  - Simple and fast to compute.
  - Works well for straightforward comparisons.
- **Limitations**:
  - Sensitive to distortions or shifts in the sequences.
  - Ignores structural alignment and relationships.

---

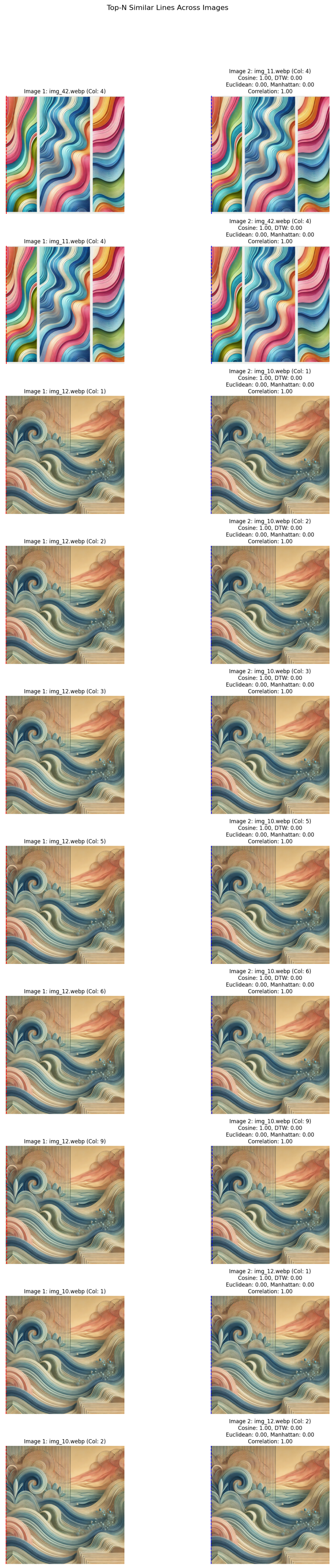
4. Correlation

- **Definition**:
  - Measures the strength and direction of the linear relationship between two sequences (e.g., pixel intensities).
- **Advantages**:
  - Identifies proportional changes in pixel intensity between two vertical lines.
  - Effective when patterns are linearly related.

- **Limitations**:
  - Less robust for nonlinear relationships or distortions.

Process and Visualization

1. **Extraction**:
   - Extract all vertical lines (columns) from the images.
2. **Comparison**:
   - Compute similarity scores for each pair of vertical lines across images using the chosen metrics.
3. **Ranking**:
   - Rank pairs based on their similarity scores.
4. **Visualization**:
   - Highlight the most similar vertical lines in the images.
   - Use heatmaps or annotated images to display results.


Top-N Similar Lines Across Images
Image 1: img_42.webp (Col: 4)
Image 2: img_11.webp (Col: 4)
Cosine: 1.00, DTW: 0.00
Euclidean: 0.00, Manhattan: 0.00
Correlation: 1.00
Image 1: img_11.webp (Col: 4)
Image 2: img_42.webp (Col: 4)
Cosine: 1.00, DTW: 0.00
Euclidean: 0.00, Manhattan: 0.00
Correlation: 1.00
Image 1: img_12.webp (Col: 1)
Image 2: img_10.webp (Col: 1)
Cosine: 1.00, DTW: 0.00
Euclidean: 0.00, Manhattan: 0.00
Correlation: 1.00
Image 1: img_12.webp (Col: 2)
Image 2: img_10.webp (Col: 2)
Cosine: 1.00, DTW: 0.00
Euclidean: 0.00, Manhattan: 0.00
Correlation: 1.00
Image 1: img_12.webp (Col: 3)
Image 2: img_10.webp (Col: 3)
Cosine: 1.00, DTW: 0.00
Euclidean: 0.00, Manhattan: 0.00
Correlation: 1.00
Image 1: img_12.webp (Col: 5)
Image 2: img_10.webp (Col: 5)
Cosine: 1.00, DTW: 0.00
Euclidean: 0.00, Manhattan: 0.00
Correlation: 1.00
Image 1: img_12.webp (Col: 6)
Image 2: img_10.webp (Col: 6)
Cosine: 1.00, DTW: 0.00
Euclidean: 0.00, Manhattan: 0.00
Correlation: 1.00
Image 1: img_12.webp (Col: 9)
Image 2: img_10.webp (Col: 9)
Cosine: 1.00, DTW: 0.00
Euclidean: 0.00, Manhattan: 0.00
Correlation: 1.00
Image 1: img_10.webp (Col: 1)
Image 2: img_12.webp (Col: 1)
Cosine: 1.00, DTW: 0.00
Euclidean: 0.00, Manhattan: 0.00
Correlation: 1.00
Image 1: img_10.webp (Col: 2)
Image 2: img_12.webp (Col: 2)
Cosine: 1.00, DTW: 0.00
Euclidean: 0.00, Manhattan: 0.00
Correlation: 1.00

b)Band of lines:

- A band consists of several consecutive vertical lines grouped together (e.g., 2px, 5px, 10px, etc.).
- Each band is treated as a single entity for comparison.

Aggregation of Band Features:

- **Convert Bands into Single Representations**:
    - Aggregate the pixel values across the band (e.g., by averaging, summing, or concatenating values).
    - Each band becomes a 1D vector or a sequence of average RGB values.
- **Direct Distance Across Pixels**:
    - Treat the band as a multi-dimensional sequence and compute similarity across the entire band simultaneously.

Apply Similarity Measures:

- Similarity measures (cosine similarity, DTW, Euclidean distance, etc.) are extended to compare aggregated features or sequences of bands.

Experiment with Band Widths:

- Test different band widths (e.g., 2px, 5px, 10px, 20px) and compare results.
- Wider bands capture more global patterns, while narrower bands focus on finer details.

Band-Width Evaluation

1. Narrow Bands (2px, 5px)

- **Characteristics**:
    - Focus on fine-grained details.
    - More sensitive to local variations (e.g., texture changes, noise).
- **Advantages**:
    - Ideal for detecting subtle differences in images with fine patterns.
    - Captures local color and texture variations effectively.
- **Limitations**:

    - Sensitive to noise, making comparisons less robust for real-world images.

2. Moderate Bands (10px, 20px)

- **Characteristics**:
    - Balance between local detail and global context.
    - Capture meaningful patterns without being overly sensitive to minor distortions.
- **Advantages**:
    - Provide a good trade-off between accuracy and robustness.
    - Effective for most practical applications, such as finding similar textures or regions.
- **Limitations**:
    - May miss very small or subtle differences.

3. Wide Bands (50px and beyond)

- **Characteristics**:
    - Capture global patterns and general color distributions.
    - Less sensitive to local variations.
- **Advantages**:
    - Robust to noise and minor distortions.
    - Suitable for detecting large-scale similarities, such as overall image themes.
- **Limitations**:
    - May overlook fine details or small differences between images.
    - Computation becomes heavier as the width increases.

Comparing Bands

1. Aggregating Band Features

- **Averaging**:
    - Compute the mean pixel value for each column within the band.
    - Simple and effective for reducing dimensionality.

- **Summing**:
  - Sum pixel intensities for each column.
  - Preserves overall intensity patterns but can distort relative relationships.
- **Concatenating**:
  - Treat all columns as a single sequence.
  - Preserves all details but increases computational complexity.

2. Simultaneous Distance Across Bands

- **Dynamic Time Warping (DTW)**:
  - Compare entire bands by aligning sequences along both the horizontal and vertical dimensions.
  - Effective for handling shifts or distortions across the band.
- **Cosine Similarity**:
  - Compute the angular similarity of the aggregated feature vectors.
  - Robust to intensity scaling, especially for color-based comparisons.
- **Euclidean and Manhattan Distances**:
  - Measure pixel-wise differences directly.
  - Simple but sensitive to distortions.

Output: Kindly refer the code, since the output visualization is huge, i didn’t include it here

Conclusion

- **Moderate Band Widths (10px to 20px)**:
  - Typically deliver the most credible results, balancing fine details and global patterns.
- **Wider Bands (>50px)**:
  - Better for general similarities but may overlook finer details.
- **Narrow Bands (2px to 5px)**:

    - Best for applications requiring high sensitivity to local changes but are more noise-prone.

C) Effects of Resizing the images

Resizing/Reshaping Scenarios

1. **Uniform Scaling**:
    - Scaling both width and height by the same factor (e.g., 50%, 200%).
    - Maintains the aspect ratio of the image.
2. **Non-Uniform Scaling (Stretching)**:
    - Scaling width and height by different factors (e.g., width 50%, height 200%).
    - Changes the aspect ratio, introducing distortions.
3. **Extreme Resizing**:
    - Downscaling to very small dimensions (e.g., 10×10) or upscaling to very large dimensions.

1. Uniform Scaling

- **Effect on Single-Line or Band Comparisons**:
    - Single lines and narrow bands are less affected as they maintain their relative proportions.
    - Wider bands may aggregate over a larger or smaller set of pixels, slightly altering the similarity score.
- **Tolerance**:
    - Methods like cosine similarity and DTW are generally robust to uniform scaling, especially for color patterns.
    - Euclidean and Manhattan distances may show more variance due to changes in magnitude.

2. Non-Uniform Scaling (Stretching)

- **Effect on Single-Line or Band Comparisons**:

- Alters the aspect ratio, stretching or compressing vertical lines or bands.
- May introduce more significant distortions in the extracted bands, especially for methods sensitive to spatial arrangement.

- **Tolerance**:
  - DTW can handle such distortions better than simple distance metrics like Euclidean.
  - Cosine similarity might show lower scores due to changes in the angular relationship between vectors.

3. Extreme Resizing

- **Downscaling**:
  - Significant loss of detail, merging multiple lines or bands into a single pixel value.
  - Metrics may fail to capture meaningful similarities.
- **Upscaling**:
  - Interpolation introduces artificial pixel values, which can distort the true patterns.
  - Similarity scores may fluctuate, especially for methods dependent on exact pixel values.

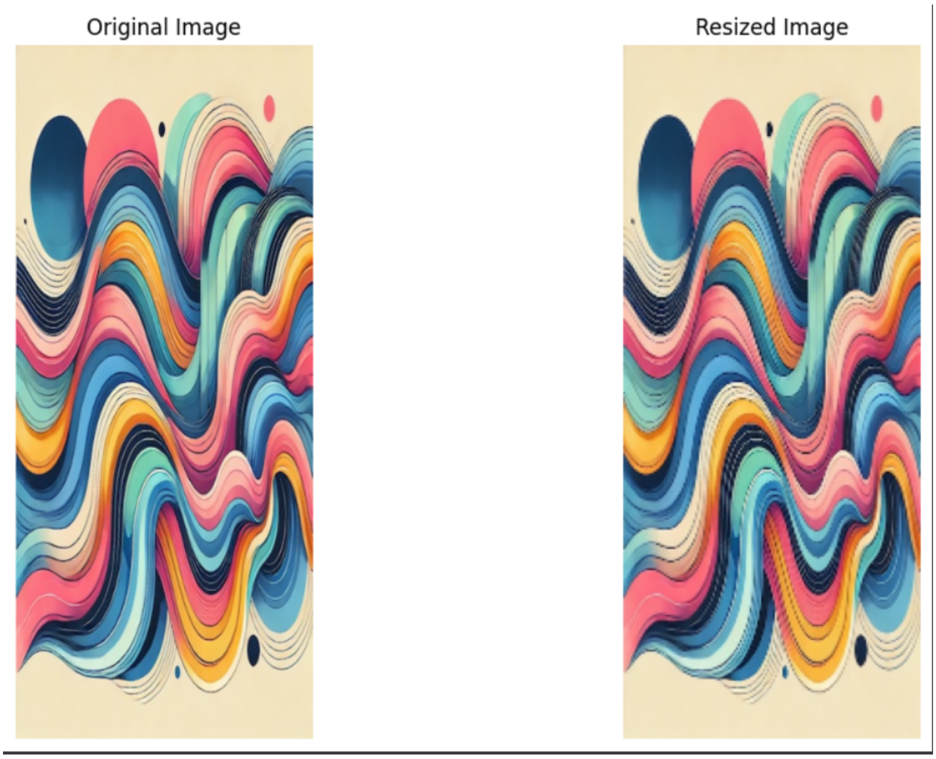
Resizing affects the performance of single-line and multi-pixel band comparisons differently:

- **Uniform scaling** introduces minimal changes, with most metrics remaining reliable.
- **Non-uniform scaling** can distort patterns, requiring robust methods like DTW.
- **Downscaling** significantly reduces detail, and **upscaling** adds artificial artifacts.

Output:

I have included only one scale due to space issue, kindly refer the code output

Testing for scale_x=0.5, scale_y=0.5



Similarity results for bands (band width: 10):
Band 0: Cosine=0.95, DTW=895.16
Band 1: Cosine=0.87, DTW=1257.44
Band 2: Cosine=0.79, DTW=1019.35
Band 3: Cosine=0.89, DTW=1091.00
Band 4: Cosine=0.88, DTW=1497.25
Band 5: Cosine=0.83, DTW=1487.35
Band 6: Cosine=0.86, DTW=1475.25
Band 7: Cosine=0.90, DTW=1000.84
Band 8: Cosine=0.93, DTW=1167.94
Band 9: Cosine=0.90, DTW=1353.83
Band 10: Cosine=0.89, DTW=1198.19
Band 11: Cosine=0.88, DTW=1248.45
Band 12: Cosine=0.87, DTW=1106.53
Band 13: Cosine=0.90, DTW=1276.26
Band 14: Cosine=0.90, DTW=1138.39
Band 15: Cosine=0.89, DTW=3228.02